An Energy-Efficient Neural Network Accelerator with Improved Protections Against Fault-Attacks

Saurav Maji*, Kyungmi Lee*, Cheng Gongye†, Yunsi Fei† and Anantha P. Chandrakasan*
*Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA
†Dept. of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA

ESSCIRC 2023- IEEE 49th European Solid State Circuits Conference (ESSCIRC) | 979-8-3503-0420-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ESSCIRC59616.2023.10268746

***Abstract*—Embedded neural network (NN) implementations are susceptible to misclassification under fault attacks. Injecting strong electromagnetic (EM) pulses is a non-invasive yet detrimental attack that affects the NN operations by (*i*) causing faults in the NN model/inputs while being read by the NN computation unit, and (*ii*) corrupting NN computations results to cause misclassification eventually. This paper presents the first ASIC demonstration of an energy-efficient NN accelerator with inbuilt fault attack detection. We incorporated lightweight cryptography-aided checks using the *Craft* cipher for on-chip verification to detect model/input errors and also as a fault detection sensor. Our developed ASIC has demonstrated excellent error detection capabilities (100% detection for 100*k* error attempts) with a minimal area overhead of 5.9% and negligible NN accuracy degradation.**

I. Introduction

There are increased security concerns with the popularity of embedded NN implementations [1]. Physical attacks over NNs can be broadly categorized as: *(i) side-channel attacks* (SCAs) that are aimed to extract confidential information [2], [3]; and *(ii) fault attacks* (FAs) that are targeted to make them dysfunctional [4]. While previous works have focused on the SCAs of NNs [5], [6], less attention has been given to FAs. In FAs, attackers introduce faults in the execution of the target platform by feeding faulty data or operating it outside normal conditions. FAs for NNs have been demonstrated for targeted misclassification and denial-of-service attacks [4], whereby attackers inject erroneous operations by manipulating the supplied voltage, altering the clock signal frequency, applying strong EM pulses, or laser pulses [7]. Among these FA methods, clock/voltage attacks require direct access to the hardware, while laser-based FAs are non-invasive and require the decapsulation of the ICs. Hence in this work, we consider electromagnetic fault injection (EMFI) attacks which are non-invasive and do not require any decapsulation.

In typical embedded NN accelerators, the NN model is stored externally (e.g., in DRAM). In this use case, an attacker can inject errors in two ways: *(i)* by introducing errors during the transmission of the NN model/input to the ASIC, and *(ii)* by causing computation errors during NN processing. Our designed ASIC possesses high error detection capabilities for both these attacks while incurring low-performance overheads.

This project was supported partly by Analog Devices Inc. and partly by MIT EECS MathWorks Fellowship. C. Gongye acknowledges National Science Foundation (NSF-SaTC 1929300) for funding support.

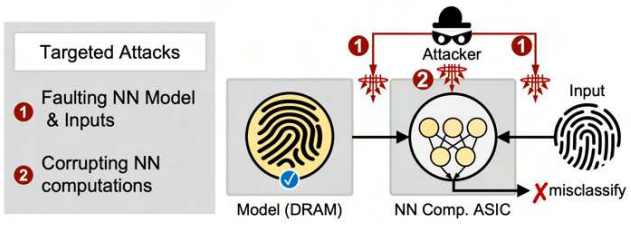


Fig. 1. Overview of EM fault attacks on embedded NNs (exemplified for thumbprint-based biometric authentication).

II. Application Scenario & Threat Model

For demonstrating EMFI over real-life applications, we consider a smart-card application that uses embedded NNs to authenticate users based on their biometric thumbprint information, as shown in Fig. 1. The user enters their thumbprint in the smart-card reader, which processes the input thumbprint to extract refined features and transmits them to the smart-card processor for verification. The smart-card processor (i.e., the NN accelerator) reads the NN model from DRAM and performs NN processing over the received features to authenticate the user. An attacker can inject faults during transmitting input features, reading the model, or during NN computations to cause targeted misclassification and perform denial-of-service attacks. We assume a white box scenario for the attack, where the attacker has extensively characterized the system to inject faults effectively. An adversary can use EM fault attacks that can be applied to both contact and contactless modes of interaction between the smart-card and the smart-card reader.

Our baseline NN accelerator comprises 16 processing elements (PEs) and supports multi-layer perceptron and convolutional NN operations. The baseline NN accelerator uses quantized 8*b* unsigned integer weights 8*b* signed activations. Fig. 2 shows the architecture of our ASIC, which augments the baseline accelerator with *(i)* lightweight *Craft* [8] cipher-aided NN model/input authentication unit and *(ii) Craft* cipher-based clock-glitch detection units. If errors are detected, the system can be configured to repeat the erroneous operations for accurate NN processing, such as requesting to re-transmit the faulty NN model/input or repeating the inaccurate NN computations. In case repeated errors are detected, the system/user is alerted regarding an attacker's presence.

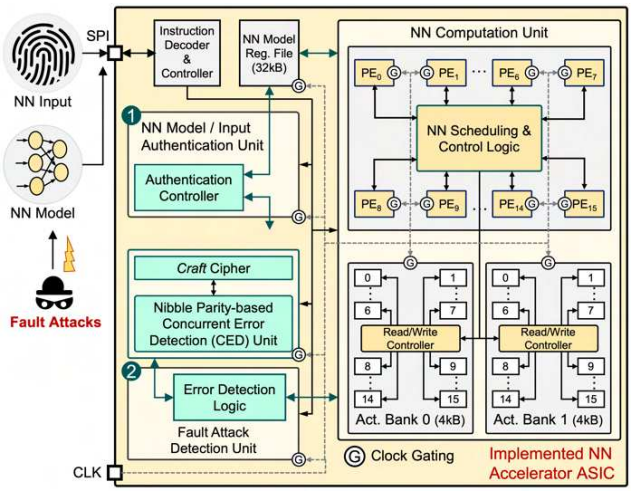


Fig. 2. System architecture of the implemented NN accelerator.

III. NN Model / Input Authentication

Current data authentication techniques include parity checks [9], algorithmic [10], or cryptographic [11] checksums. The parity/algorithmic checksums require very low memory overheads but do not perform well when several bits can be flipped arbitrarily (e.g., flipping even bits for parity checks). On the contrary, cryptographic checks offer very strong security guarantees. However, they lead to high performance overheads. As an example, incorporating 64b cryptographic checksum into the entire NN model will cause negligible memory overhead. On the mismatch of the checksum, the entire model data needs to be re-fetched, thus leading to high data transmission overhead. On the other hand, incorporating cryptographic checksums at smaller memory intervals helps to locate memory errors at smaller granularity. However, it requires high memory overheads. For example, 64b cryptographic checksum for 0.5kB memory leads to 12.5% memory overhead. Our proposed method, which we discuss next, incorporates parity checks under the umbrella of encryption, thus combining the benefits of both these methods.

As shown in Fig. 3, our proposed method's parity check is inspired from [9], which proposes to set the least significant bit (LSB) of each weight as its corresponding even parity bit. This is based on the fact that the LSBs contribute very less to the NN accuracy. Additionally, we also encrypt the parity-encoded model. Therefore, the model verification involves on-chip decryption of model parameters and performing parity checks. We partition the NN model into 0.5kB blocks (64×8b weights) and process them individually. A block is detected as valid when parity checks for all its 64 weights are satisfied. Without the knowledge of the cipher key, it is extremely challenging for an attacker to alter the encrypted model such that the decrypted model bypasses the checks. We also organized the data so that all equipositional bits (e.g., LSBs) of all weights are grouped in a row and encrypted. Thus, any anomaly in the encrypted block possibly affects all weights. Hence, for any altered ciphertext, each of the 64 weights (in the plaintext) satisfies the parity condition with a probability of 0.5 (i.e., random guess). Hence, the probability of a fault-injected block (with 64 weights) passing the check has an insignificant probability of $(0.5)^{64} \simeq 0\%$. As exemplified in Fig. 4, flipping even a single bit in the ciphertext dismantles the parity for several weights in the resultant plaintext.

The methodology that is described above and used for anomaly detection in NN model, is also used for verifying the NN input. This leads to effective resource utilization.

We utilized *Craft* [8], a lightweight tweakable block cipher, for our application. Tweakable ciphers are a special category of block cipher which are popular for encrypting memory. They provide an efficient way to encrypt an entire memory with a single key while ensuring that different memory addresses (tweaks) have uncorrelated encryptions. We opted for *Craft* instead of existing solutions such as XTS [12] for two reasons: *Craft* being a lightweight cipher is more efficient and uses fewer resources than AES, and the design of *Craft* allows for easy error detection capabilities (that is utilized in Section IV). In Section V, we discuss the impact of enforcing parity constraints on the accuracy of our NN. While our proposed method does not increase the size of the model/input, we do need to pad the model/input to make it a multiple of the block size, which results in negligible memory overheads of <0.5kB.

IV. Digital Sensor for NN Fault Detection

Cryptographic ciphers are highly sensitive to fault attacks. Hence, any coarse-grained EM attack that is targeted over the NN is likely to affect the cipher, thus corrupting its state. Hence, we detect glitches using the cipher's inherent vulnerability. We complement the *Craft* cipher with a nibble (4b)-parity-based concurrent error detection (CED) unit [8]. An anomaly between the cipher's state and its parity indicates an error. The cipher and its CED unit execute concurrently with the NN computation and raise an error flag when obtaining a parity mismatch, as shown in Fig. 5. Our proposed method is capable of detecting NN computation errors for all PEs without incorporating dedicated operation-specific verification (e.g., separate checks for addition-&-multiply (MAC), ReLU, etc.) or redundant NN computations. Our method is fully synthesizable and does not require technology-specific design/characterization required for designing sensors [13], [14]. Because of the effective reuse of the existing cipher (from Section IV), our solution incurs low area overhead while providing high error detection capabilities.

V. Experimental Demonstration of Fault Attack Resistance For Bio-metric Authentication

Our designed ASIC was used to demonstrate the real-life application of thumbprint information-based biometric authentication [15]. Fig. 6 shows the NN model architecture. As demonstrated here and analyzed in [9], the degradation in the NN accuracy is extremely small (almost negligible) while enforcing the parity constraints for the weights and inputs.

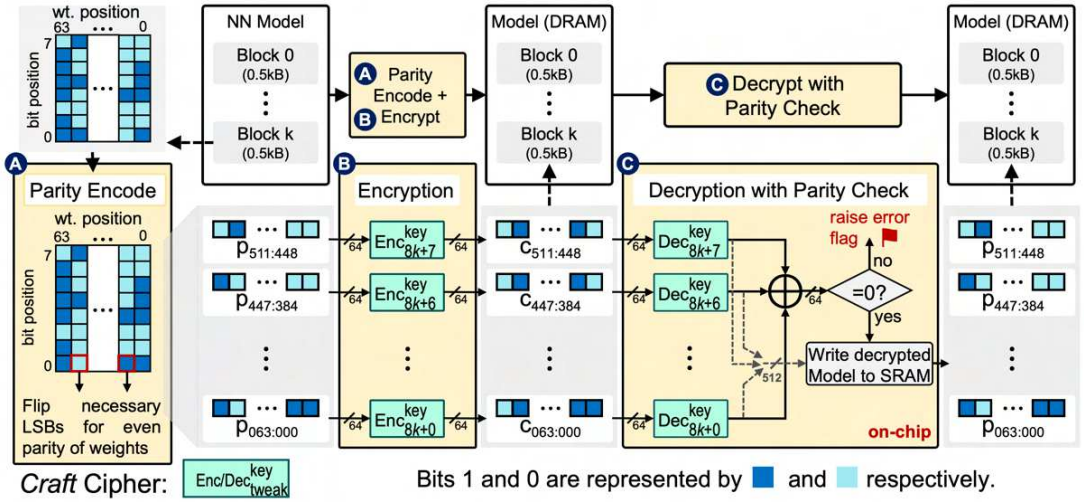


Fig. 3. Proposed NN model authentication: algorithm and implementation.

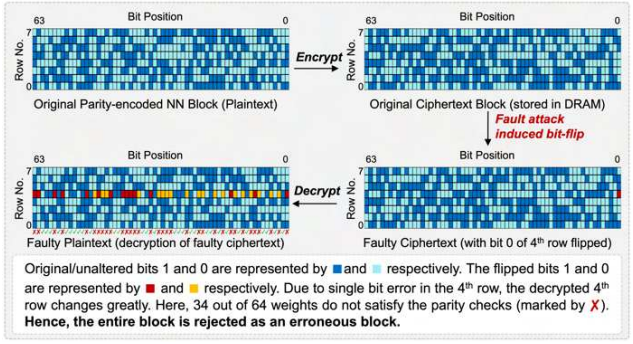


Fig. 4. Illustrated example of NN model authentication.

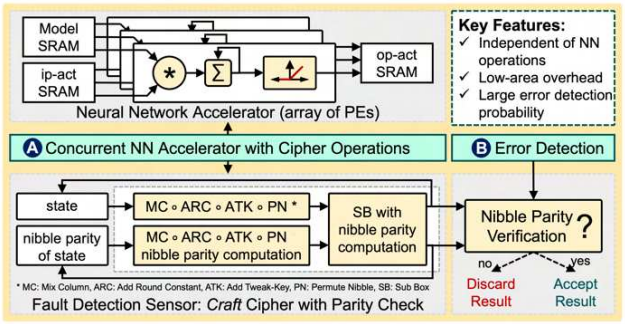


Fig. 5. Design of the cipher-aided fault detection unit for NN computations.

For the EM glitch attack, we utilized Riscure's EMFI injection probe, with a diameter of 4mm. To carry out an effective fault attack, we pinpointed the most vulnerable position by deactivating protections (using exhaustive search). We subsequently activated the implemented protections and analyzed the response against EM fault attacks.

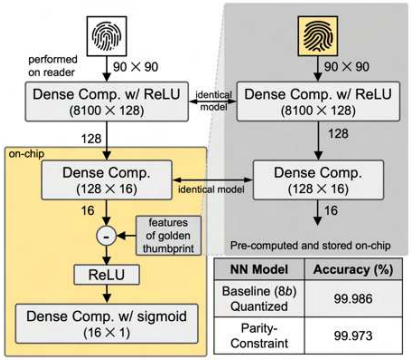


Fig. 6. Model details and performance of the demonstrated application of thumbprint-based user recognition.

TABLE I
SUMMARY OF FAULT-ATTACK EXPERIMENTS

Fault attacks	No. of attacks	Fault Coverage
NN Model	0.1M	(0.1M/0.1M) = 100%
NN Input	0.1M	(0.1M/0.1M) = 100%
NN Computations	0.1M	(0.1M/0.1M) = 100%

The conducted experiments are described below and summarized in TABLE I :

- For our demonstrated applications, we introduced errors into the NN inputs and the NN model (particularly biases as they are most sensitive for accuracy) and achieved misclassification. However, in the protected implementation, any errors introduced over the NN model were detected (for $100k$ FA attempts).
- We injected random errors during the NN computations and were able to misclassify the output. However, in the protected implementation, any error injected was detected correctly for $100k$ attempts.

TABLE II
COMPARISON WITH PRIOR WORKS ON PROTECTION AGAINST ACTIVE ATTACKS.

Metric	ISSCC'11 [16]	JSSC'18 [13]	VLSIC'22 [14]	ISSCC'23 [17]	This Work
Platform	ASIC (130nm)	ASIC (180nm)	ASIC (5nm)	ASIC (Intel 4)	ASIC (28nm)
Application	AES	AES	–	AES	Neural Networks
Targeted Attacks	Any FA	Laser	Clock Glitch	Any FA	EM Faults
Defense Techniques	Duplicate datapath	Bulk current sensors	FLL-based detector	Arith. + Parity Checks + LDC	Cipher-based anomaly detection
Supply Voltage (V)	1.20	1.80	0.75	0.75	0.60 – 0.95
Frequency (MHz)	50	25	40	780	25 – 200
Energy/opn.	–	0.6 nJ/cycle	20.1 pJ/cycle [a]	34.6 pJ/cycle [a]	1.37 pJ/MAC (unprotected)[b] 1.88 pJ/MAC (protected)[b]
Energy Overhead	–	0.3%	20.1 pJ/cycle [c]	–	37%
Area Overhead	104%	28%	0.0048 mm^2 [c]	40%	5.9%

[a] Computed from the reported power & frequency. [b] Reported at 0.60V & 25MHz. [c] Constant energy & area overhead.

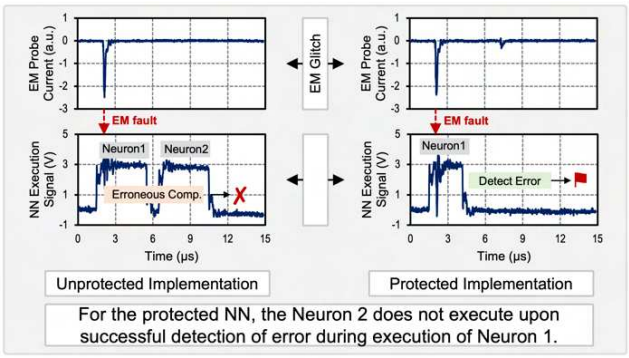


Fig. 7. Successful demonstration of EM glitch detection capability of the implemented NN accelerator.

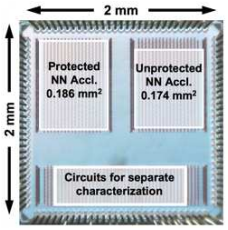
As shown in Fig. 7, for an unprotected implementation, the error induced during the execution of neuron 1 propagates to neuron 2 and gives incorrect decision. Whereas, for the protected implementation, an error flag is raised on successfully detecting fault and the neuron 2 does not execute.

VI. Comparison With Other Fault Tolerant Designs

Fig. 8 shows the fabricated ASIC (in 28nm HPC+ CMOS) and summarizes its performance. Our chip supports voltage scaling from 0.95V to 0.60V and operates from 25MHz to 200MHz. All the energy measurements have been reported at 0.60V and 25MHz. TABLE II compares our work with prior work on custom hardware designs for fault attack-resistant ASICs. These prior works have targeted fault-tolerant applications for cryptographic applications. Our work is the only one to target FAs for NN applications while achieving high error-detection capabilities and low area overheads. Using algorithmic and architectural innovations, this work demonstrate the first NN accelerator ASIC resilient against fault attacks for resource-constraint applications.

Acknowledgments

The authors gratefully acknowledge the TSMC University Shuttle Program for providing chip fabrication support.



Specifications of the Protected Neural Network Accelerator	
Logic Area	70.04 kGE
Memory	40 kB
Operating Frequency	25MHz (0.60V) 200MHz (0.95V)
Power	1.25mW (0.60V, 25MHz) 10.1mW (0.95V, 200MHz)
No. of PEs	16

Fig. 8. Chip micrograph and specifications.

References

[1] Q. Xu *et al.*, "Security of Neural Networks from Hardware Perspective: A Survey and Beyond," in *ASP-DAC*, 2021.
[2] L. Batina *et al.*, "CSI NN: Reverse Engineering of Neural Network Architectures Through Electromagnetic Side Channel," in *28th USENIX Security Symposium*, 2019.
[3] S. Maji *et al.*, "Leaky Nets: Recovering Embedded Neural Network Models and Inputs Through Simple Power and Timing Side-Channels—Attacks and Defenses," *IEEE JIoT*, vol. 8, no. 15, 2021.
[4] S. Tajik *et al.*, "Artificial Neural Networks and Fault Injection Attacks," *arXiv preprint arXiv:2008.07072*, 2020.
[5] S. Maji *et al.*, "A ThresholD-ImpIementation-Based Neural-Network Accelerator Securing Model Parameters and Inputs Against Power Side-Channel Attacks," in *ISSCC*, 2022.
[6] S. Maji *et al.*, "A Threshold Implementation-Based Neural Network Accelerator With Power and Electromagnetic Side-Channel Countermeasures," *IEEE JSSC*, vol. 58, no. 1, 2023.
[7] J. Breier *et al.*, "How Practical Are Fault Injection Attacks, Really?" *IEEE Access*, vol. 10, 2022.
[8] C. Beierle *et al.*, "CRAFT: Lightweight Tweakable Block Cipher with Efficient Protection Against DFA Attacks," *IACR ToSC*, 2019.
[9] S. Burel *et al.*, "Zero-Overhead Protection for CNN Weights," in *IEEE DFT*, 2021.
[10] J. Li *et al.*, "RADAR: Run-time Adversarial Weight Attack Detection and Accuracy Recovery," in *DATE*, 2021.
[11] R. Elbaz *et al.*, "Hardware Mechanisms for Memory Authentication: A Survey of Existing Techniques and Engines," *LNCS*, vol. 5430, 2009.
[12] L. Martin, "XTS: A Mode of AES for Encrypting Hard Disks," *IEEE Security & Privacy*, no. 3, 2010.
[13] K. Matsuda *et al.*, "A 286 F^2/Cell Distributed Bulk-Current Sensor and Secure Flush Code Eraser Against Laser Fault Injection Attack on Cryptographic Processor," *IEEE JSSC*, vol. 53, no. 11, 2018.
[14] S. Song *et al.*, "An FLL-Based Clock Glitch Detector for Security Circuits in a 5nm FINFET Process," in *VLSI*, 2022.
[15] Y. I. Shehu *et al.*, "Sokoto Coventry Fingerprint Dataset," *arXiv preprint arXiv:1807.10609*, 2018.
[16] M. D.-Verdier *et al.*, "A Side-Channel and Fault-Attack Resistant AES Circuit Working on Duplicated Complemented Values," in *ISSCC*, 2011.
[17] R. Kumar *et al.*, "A 100Gbps Fault-Injection Attack Resistant AES-256 Engine with 99.1-to-99.99% Error Coverage in Intel 4 CMOS," in *ISSCC*, 2023.